

To Do or not to Do Previous Homework in Computer Education

Sónia Rolland Sobral
REMIT
Universidade Portucalense
 Porto, Portugal
 sonia@upt.pt

Abstract—The first contact of freshmen students with computer thinking and programming languages is not an easy task. There are several strategies that can be used before, during and after face-to-face classes. Flipped classes are a way to save time in activities that can be better done in the form of previous work - guided by the teacher. And the time gained in person can be used in tasks that are much more productive than traditional lectures.

This study is based on an introductory programming semester of post lockdown COVID-19, initially with 101 students, with the strategy of providing study materials for students to work on before class (as in flipped classes). We use attendance in classes, access-work MOODLE, the two grades obtained during the semester, as well as some information about the student (age, course, gender, previous knowledge of programming languages), and the level that the students think it was their presence in classes and on MOODLE in the middle and at the end of the semester to measure the success of the experiment. It seems that this type of strategy can be excellent for students who attend classes weekly and do their homework, but it can be a cause of dropout if taken to the extreme.

Keywords—programming fundamentals, novice students, flipped classroom

I. INTRODUCTION

Flipped classroom learning strategy is based on the dynamics of reversing the traditional teaching-learning order: students study new materials before class, leaving class time reserved for discussion, questioning, and application of previously acquired knowledge. This way, students can learn and work previously at their own pace and have more quality time with teacher support. Teachers can solve more interesting exercises during classes, not wasting time with classes to transmit knowledge that is often very theoretical and needs some time to understand. The flipped classroom concept aims to reverse the way content is taught and applied. Traditionally, content is presented in face-to-face classes and homework is done afterwards, in the flipped class strategy, homework is done before class, turning classroom work much more productive. This educational technique is then divided into two parts: direct individual computer-based instruction outside the classroom and interactive group learning activities within the classroom [1]. Flipped classrooms are intended to promote the improvement of learning outcomes such as student performance, achievement and engagement This teaching-learning method has been used in several areas [2] [3] [4] and different levels of education [5] [6].

Learning management systems (LMS), distance learning enabling platforms such as MOODLE, have been used around the world for many years. During the COVID-19 lockdown, LMS played a very important role, along with video

conferencing tools: turning emergency remote teaching possible. The advantages of these platforms are clear: they facilitate the availability of study materials to students who can study at their own pace (when and where). LMS are a great way to organize homework (also before-classroom): teacher sends the next week's material, indicating what materials are needed for homework. These materials accessible from the LMS improve the organization of the study – the student feels accompanied and guided in their work. The LMS are also an excellent tool for the Teacher – not only in pedagogical terms, but because they allow them to know a lot of information about the students' interaction with the study, allowing them to activate tools for the prevention of failures and dropouts.

Programming fundamentals courses has been the concern of many researchers [7] [8]. In these types of subjects, students are introduced to the world of programming languages and computational thinking. The failure and dropout rates mean that teachers have a constant search for methods and strategies that can work with this type of subjects [9] [10]. Flipped lessons has been widely used in the context of introductory programming [11] [12] [13]. The first experiences were reported ten years ago [14] [15] [16], since then many more have followed [17] [18] [19].

The experience that this article reports took place during a semester in a first-year curricular unit common to three different degree courses in informatics. At the beginning of the semester, students were asked to answer an initial questionnaire, the same happening in the middle of the semester and at the end. 79 of the 101 students enrolled in the course responded to the first survey. During the semester, we tried to use the strategy of flipped classes: the teacher sent an email on Friday, summarizing the week and presenting which materials were in MOODLE and which would be worked on following week. This article relates the consultation of materials before classes with the classifications obtained.

This article is divided into literature review, methodology, results and discussion of results and conclusions.

II. LITERATURE REVIEW

Bishop and Verleger [1] define the flipped classroom as an educational technique divided into two parts: 1. Direct individual computer-based instruction outside the classroom and 2. Group learning activities within the classroom. Sobral [13] writes that the dynamics of this teaching-learning technique is to invert the traditional order, with students having contact with new subjects before the classroom, and face-to-face classes are reserved for application of previously acquired knowledge. Long and colleagues [20] characterize the flipped classroom model as an instructional model in which students learn basic subject matter knowledge before

classroom meetings, then come to the classroom for active learning experiences.

There are studies on inverted classes from all over the world. Saeed and Al-Emran [21] found further investigations in China, Turkey, Taiwan and Malaysia; Sobral [13] found mainly in the United States and Canada, Julia and colleagues [22] in the United States of America, China, Taiwan, Australia and Indonesia.

In the literature we find many references to flipped classes, most of them in the context of higher education [23], although there are some studies related to k-12 [24]. Most of the flipped courses were related to the STEM (Science, Technology, Engineering e Mathematics) field [24] but also in the health area [25] [26] [3] and others like English language teaching [2].

Introductory programming is the most common name for courses found by in the context of higher education and teaching programming using the dynamics of flipped classes [12] using several programming languages, being Java the adopted programming language that appears more [13]. This finding agrees with what has already been reported about introductory programming courses and the languages that are currently most adopted (java, C and Python) [27] [28].

In the literature we find different experiences that report different results. Most articles compare the use of inverted classes with traditional classes [29] [30]. There are courses that are implemented exclusively using the strategy of inverted classes [31] [32], while others use this method only in some subjects or classes [33]. In the literature we find many objectives for studies and experiences in programming teaching and inverted classrooms: most are concerned with programming self-efficacy and academic performance, but there are other objectives like acceptance by gender

The flipped classroom strategy is generally divided into two steps: outside the classroom and inside the classroom. In the study of Alhazbi [34], we found a definition with a third stage, the in-lab activities, used to develop students, in a practical way, problem analysis skills and algorithm development to create new programs.

The technology used for teachers to make materials available for students to access before classes is diverse: Learning management systems are widely adopted for the ease of integration of activities, namely MOODLE [35], blackboard [36] and Coursera MOOCs [37]. But there are also those who choose to use "off-campus" technology, such as Facebook [38], WordPress [39] or on the teacher's YouTube channel [40]-

The most used activities outside the classroom are videos, quizzes, readings, online modules, self-exercise, assignments, and interactive tutorials [12]. There are reports of experiences in which books are replaced by texts in their online version to be read before classes "replacing the traditional textbook and online tutoring system with an integrative version" [41] or Microsoft PowerPoint slides [42]. The videos are usually made available by indicating the link of the videos on YouTube to materials prepared by the teacher [43] or made by other people or companies [44]. There is an interesting strategy of proposing to students to look for videos that fit the subject in question [45]. There are many video strategies: some videos are one hour long [29], others are short videos (7 to 11 minutes, for example) [46]. Regarding quizzes, there are

also different understandings: before classes [44] or in the first minutes in the classroom [47].

According to Hendrik and A. Hamzah [12], the most reported activities in the literature are hands-on-experience, small briefings, quizzes, assignments with teacher assistance, and questions and answers. There are pedagogical techniques that are used in-class, such as Pair Programming [48], Project based learning [49], Collaborative learning [50], Think-Pair-Share [51], Peer Instruction [52], Team-Based Learning [53], and Problem-Based Learning [54].

III. METHODOGY

The curricular unit Algorithms and Programming is mandatory and exists in the first semester of the degree courses in computer science, computer engineering and industrial engineering and management. Weekly there are two theoretical hours and four (or two) laboratory hours on the university campus. The programming language adopted is Python to code small problems. There is a great emphasis on computational thinking, using various tools such as top-down and algorithmic language. Assessment consists of two tests throughout the semester (45% each) and class assessment (10%), with more than 70% compulsory attendance in classes. Students only have access to the second test if they have at least 7 out of 20. This article describes continuous assessment. Students who fail in continuous assessment still can take a final exam, and in some special cases to take an exam at an additional time.

This article refers to a shorter academic semester than usual due to contingencies of the COVID-19 pandemic - taking 12 weeks and not 15 weeks as previous academic years. Every week, on Friday, an announcement is sent with a summary of the previous week and the materials for the following week. The platform used is MOODLE. The materials are placed weekly on the curricular unit page, as well as the proposed exercises. At the request of the students, the resolution of the exercises from the previous week is also placed.

When we refer to different types of materials, we are referring to different types of materials, such as pdf lectures and narrated presentations. Other interactive content is proposed according to the topic of the week: links to videos or tutorials, as well as small programs in Python programming languages with explanations or small quizzes on MOODLE. We try to ensure that the number of activities per week is correct and can be carried out during the normal period.

At the beginning of the semester, students are invited to complete a survey for demographic characterization, previous programming knowledge and other elements that are not relevant to this article. Of the 101 students enrolled, 11 students never attended any classes or tests. 79 responded to the original survey.

The mid-semester survey was answered by 74 students. The end of semester survey was answered by 54 students. These two surveys are used for the student to say (1-7) what he thinks his attendance is, what he thinks his level of study of the materials is before classes and what his grade is expected in the next test.

In this article we use student demographics (gender, age group, type of student and others) and grade expectations, as well as what level of study and attendance the student considers he/she had. We also use classroom attendance

records, MOODLE activity, and first and second test scores during the continuous assessment season. The p-value (0.01) is used to verify if there are significant correlations between the variables.

IV. RESULTS

A. Initial Survey

79 students responded to the initial survey: 14 are female and 65 are male (82%).

9 from the Informatics course, 16 from Industrial Engineering and Management and 54 from Informatics Engineering (68%).

61 students are 21 years old or younger (77%), 18 students are 21 years old or older.

42 are Ordinary Student (53%), 14 students who have a timetable overlapping with other subjects in subsequent years and 23 take some curricular units separately.

86% of students say they have a laptop and 100% say they have a smartphone.

36 had no programming languages or computational thinking subjects in high school, 43 had (54%).

35 had no programming knowledge, 44 did (56%).

Most gave 5 (scale 1-5) to the statement "I like technology": mode 5 (49), mean 4.56, standard deviation 0.63.

Most gave 5 (scale 1-5) to the statement "I am interested in technology": mode 5 (52), mean 4.58, standard deviation 0.67.

Most assigned 4 (scale 1- 5) to the statement "I have technology skills": mode 4 (43), mean 3.58, standard deviation 0.86.

Most assigned 4 (scale 1-5) to the statement "I am motivated to learn technological skills": Mode 4 (37), mean 4.02, standard deviation 0.7.

B. Attendance to Face-to-face Classes

Attendance is up to date in the university system. A regular student must attend at least 70% of classes to pass the regular season, and this obligation is not necessary for students with overlapping schedules.

In both the first and second half of the semester, 90% of normal students had more than 70% attendance, with only 1 of the overlapping students attending 70% or more classes.

When asked about their level of attendance, all respondents said approximately (difference of 2 from reality) their attendance rate (0-7) either in the middle or at the end of the semester.

Applying the p test to cross-data (previous section and this one), it is verified that there are no significant differences in relation to the variables gender, course, age group, type of student, possession of a laptop and smartphone, prior knowledge, and relationship with technology, nor a grade obtained in the first or in the second test, in terms of actual attendance, attendance that the student says, or the relationship between these two.

C. Reading Materials before Face-to-face Classes

In this subsection, we use MOODLE logs for pre-class activities as well as student responses to mid-semester and end-of-semester surveys. In both the first and second half of the semester, students were asked to respond according to the statement "I usually read materials before class", with a level of 0-none to 7-all. We calculated the differences between reality and what each student said about their reading of the materials, as shown in the following table.

In this table we find for the middle and end of the semester: the mean, mode and standard deviation (StD) of the reality of records in MOODLE (Real), what the student thinks were their records in MOODLE (Think) and the difference between Real and Think (Dif).

TABLE I. MOODLE LOGS (REAL, STUDENT THINK AND DIFFERENCE).

	Middle			End		
	Real	Think	Dif	Real	Think	Dif
Mean	4,20	4,96	-0,76	5,11	4,69	0,42
Mode	4	5	0	4	6	0
StD	1,67	1,68	1,70	1,30	1,59	1,76

From the MOODLE activity it was found that 24% of the students consulted at least 75% of the materials in the first part and 40% in the second part, on the contrary it was found that 26% and 7% respectively did not read or 25% of the proposed materials.

By the p value, we did not find significant differences in relation to gender, course, age group, type of student, possession of a laptop and smartphone, prior knowledge, and relationship with technology, even in the second test. On the contrary, for the first test grade variable, a correlation is suggested ($p=0.0012$), as can be seen in the following graph. There may be an explanation: it is that the second test can only go to students with 7 or more values, so those who do not consult the MOODLE are eliminated in the first part.

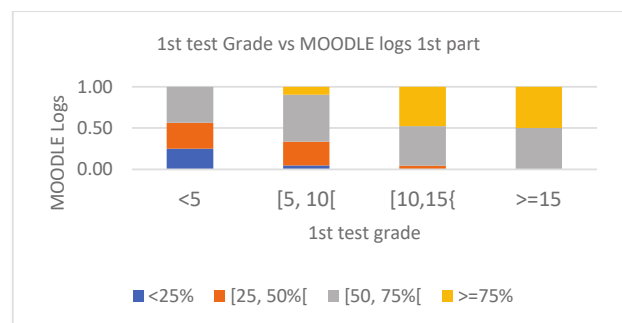


Fig. 1. 1st test Grade vs MOODLE utilization 1st part.

D. Grades

Of the students who responded to the survey mid-semester, 69 took the first test. The mean was 8.81, the lowest grade was a 1, the highest grade was a 19, the standard deviation was 5.008 and 34 grades were positive (49%). In the second test, 45 of the students who responded to the end-of-semester survey took test two. This test had scores between 1 and 19, mean of 6.81, standard deviation of 4,659 and 8 positive (18%). The results were not at all good, but they are in line with what is expected in these curricular units.

Self-assessment is a well-known strategy for engaging students. Self-assessment in this curricular unit is not used in

the calculation of the grade. However, as in other experiments [55], students were asked to predict their test scores (a first test in the middle of the semester and a second test at the end of the semester).

There is no correlation between demographics and grades. A direct relationship found between previous work on MOODLE and ratings, as mentioned in the previous section. The usual difference between grades and grade prediction is found ($p=0.001$), the less you know, the more you think you know and the more you know, the more you know what you don't know. And also, between the difference between the predicted grade and the obtained grade, based on the previous MOODLE work level. We have divided the work levels into four, as we check the schooling level of the class and verify a series of work among these variables. From what we can say that the less a student works the more the materials that are in MOODLE - in the form of previous homework - the greater the prediction error in relation to the grade, especially the grade of the first test.

V. DISCUSSION OF RESULTS AND CONCLUSIONS

The introductory programming course traditionally has low pass rates in relation to other subjects. There are new skills that students need to acquire, as well as practice to achieve goals. The period of confinement COVID-19 made students and teachers more intensively use distance learning tools such as MOODLE and ZOOM. In this way, in the post confinement, it is possible to do a more effective work, assuming these tools as a means of work known to all. The example presented is a curricular unit of the first semester and first year of three different courses in computing. Although there were 101 students enrolled, only those who respond to the surveys are considered (79 students initially, 74 in the middle of the semester and 54 at the end of the semester). From the initial survey, it appears that 82% are male, 68% have a computer engineering course, 77% are 21 years old or younger, 86% have a laptop and 100% have a smartphone, 54% had a technology discipline in teaching, 56% with some previous programming knowledge.

Flipped classes seem to be an excellent way to get students to work outside of class with the materials chosen by the Teachers and have time during face-to-face classes to consolidate knowledge, have quality time with the teachers and be able to apply what they have learned in tasks much more interesting than just listening to the teachers. The problem presented in the literature is that some of the students do not do their homework, so face-to-face classes are no longer attractive due to lack of prior reading.

In this article it is verified that students who intend to succeed in the curricular unit must have an attendance of at least 70% in face-to-face classes. Of the 101 enrolled students, 78% responded to the initial survey, 73% to the mid-semester survey and 53% to the final survey. These numbers may be related to the number of students who drop out of the course: initially and between the first and second tests.

From the MOODLE activity it was found that 24% of the students consulted at least 75% of the materials in the first part and 40% in the second part, on the contrary it was found that 26% and 7% respectively did not read or 25% of the proposed materials. 26% of students not doing their homework makes it impossible to learn as it is designed: thus, face-to-face classes become unintelligible without prior reading of materials. We found a correlation between the score on the first test and the

reading of materials in MOODLE. Such correlation is not significant in relation to the second test – probably because the students who were not used to doing their homework had already dropped out in the second part because they did not obtain the minimum grade of 7 values.

Halfway through the semester, it appears that on a scale of 1-7, the average of 4.20 for the consultation of is too small for the materials teacher that the homework is done. This average is worsened by the students' answers to the question I usually read the materials (4.96) to be superior to reality. In this way, two strategies: a) keep what was planned, and students who did not do the homework to stay momentarily or there are always lost, or b) change what was planned and explain that it should have been worked at home by the students' materials, and students who study it previously can fulfil what was requested.

At the end of the semester, and when only the students who had a first test grade equal to or greater than 7.0 remained, it was verified that the level of previous work was improved (5.11), as well as the approximation of the semester to what the student perceives as having been your level of study (4.69).

Therefore, we can expose a relation between the previous study and grades. The strategy and its consequences are then an option of the teacher and course coordination.

We managed to identify the difference between the grades also and the usual solution. This study also predicts a forecast prediction between the predicted in MOODLE and the major prediction error of classification:

At least throughout this article we can ensure that a student can work with the materials before class has a huge chance of success. Only with work can a student succeed in this type of course unit – if not they will contribute to the usual dropout and failure rates.

REFERENCES

- [1] J. Bishop e M. Verleger, "The flipped classroom: A survey of the research," em ASEE Annual Conference and Exposition, Conference Proceedings, 2013.
- [2] Z. Turan e B. Akdag-Cimen, "Flipped classroom in English language teaching: a systematic review," *Computer Assisted Language Learning*, vol. 33, n° 5-6, pp. 590-606, 2020.
- [3] A. Vanka, S. Vanka e O. Wali, "Flipped classroom in dental education: A scoping review," *European Journal of Dental Education*, vol. 24, n° 2, pp. 213-226, 2020.
- [4] V. Fornons e R. Palau, "Flipped classroom in the teaching of mathematics: A systematic review," *Education in the Knowledge Society*, vol. 22, 2021.
- [5] H. Al-Samarraie, A. Shamsuddin e A. Alzahrani, "A flipped classroom model in higher education: a review of the evidence across disciplines," *Educational Technology Research and Development*, vol. 68, n° 3, pp. 1017-1051, 2020.
- [6] A. Al-Abdullatif, "Investigating self-regulated learning and academic achievement in an eLearning environment: The case of K-12 flipped classroom," *Cogent Education*, vol. 7, n° 1, 2020.
- [7] A. Luxton-Reilly, Simon, I. Albluwi, B. A. Becker, M. Giannakos, A. N. Kumar, L. Ott, J. Paterson, M. J. Scott, J. Sheard e C. Szabo, "Introductory programming: A systematic literature review," em *ITICSE 2018 Companion: Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 20218.
- [8] R. P. Medeiros, G. L. Ramalho e T. P. Falcão, "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education," *IEEE Transactions on Education*, vol. 62, n° 2, pp. 77-90, 2019.

- [9] J. Figueiredo e F. García-Peñalvo, "Teaching and Learning Strategies for Introductory Programming in University Courses," em TEEM'21: Ninth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM'21), 2021.
- [10] S. R. Sobral, "Strategies on Teaching Introducing to Programming in Higher Education," em Trends and Applications in Information Systems and Technologies, 2021, pp. 133-150.
- [11] E. M. Bakheet e A. M. Gravell, "Would Flipped Classroom be My Approach in Teaching Computing Courses: Literature Review," em 2021 9th International Conference on Information and Education Technology, Okayama, 2021.
- [12] H. Hendrik e A. Hamzah, "Flipped Classroom In Programming Course: A Systematic Literature Review," International Journal of Emerging Technologies in Learning, vol. 16, n° 2, pp. 220-236, 2020.
- [13] S. R. Sobral, "Flipped Classrooms for Introductory Computer Programming Courses," International Journal of Information and Education Technology, vol. 11, n° 4, pp. 178-183, 2021.
- [14] A. Vihavainen, M. Luukkainen e J. Kurhila, "Multi-faceted support for MOOC in programming," em ACM Special Interest Group for Information Technology Education Conference, 2012.
- [15] K. Lockwood e R. Esselstein, "The inverted classroom and the CS curriculum," em SIGCSE '13: Proceeding of the 44th ACM technical symposium on Computer science education, 2013.
- [16] A. Amresh, A. Carberry e J. Femiani, "Evaluating the effectiveness of flipped classrooms for teaching CS1," em Frontiers in Education Conference, FIE, 2013.
- [17] M. D'Souza e P. Rodrigues, "Investigating the effectiveness of the flipped classroom in an introductory programming course," New Educational Review, vol. 40, n° 2, pp. 129-139, 2015.
- [18] S. R. Sobral, "Flipped classroom is not for every kind of student.," em 15th Annual International Technology, Education and Development Conference, 2021.
- [19] J. Elmaleh e V. Shankaraman, "Improving student learning in an introductory programming course using flipped classroom and competency framework," em IEEE Global Engineering Education Conference, EDUCON, 2017.
- [20] T. Long, J. Cummins e M. Waugh, "Use of the flipped classroom instructional model in higher education: instructors' perspectives," Journal of Computing in Higher Education, vol. 29, p. 179-200, 2017.
- [21] R. Al-Marouf e M. Al-Emran, "Research Trends in Flipped Classroom: A Systematic Review," Studies in Systems, Decision and Control, vol. 295, pp. 253-275, 2021.
- [22] J. J. i. N. Afrianti, K. Soomro, T. Supriyadi, D. Dolifah, I. Isrokatun, E. Erhamwilda e D. Ningrum, "Flipped Classroom Educational Model (2010-2019): A Bibliometric Study," European Journal of Educational Research, vol. 9, n° 4, pp. 1377-1392, 2020.
- [23] J. O'Flaherty e C. Phillips, "The use of flipped classrooms in higher education: A scoping review," The Internet and Higher Education, vol. 25, pp. 85-95, 2015.
- [24] C. Lo e K. Hew, "A critical review of flipped classroom challenges in K-12 education: possible solutions and recommendations for future research," Research and Practice in Technology Enhanced Learning, vol. 12, n° 4, 2017.
- [25] F. Chen, A. Lui e S. Martinelli, "A systematic review of the effectiveness of flipped classrooms in medical education," Medical Education, vol. 51, n° 6, pp. 585-597, 2017.
- [26] V. Betihavas, H. Bridgman, R. Kornhaber e M. Cross, "The evidence for 'flipping out': A systematic review of the flipped classroom in nursing education," Nurse Education Today, vol. 38, pp. 15-21, 2016.
- [27] S. R. Sobral, "The first programming language and freshman year in computer," em Advances in Intelligent Systems and Computing, 2020.
- [28] S. R. Sobral, "The Old Question: Which Programming Language Should We Choose to Teach to Program?," em Advances in Intelligent Systems and Computing, 2021.
- [29] D. Horton, M. Craig, J. Campbell, P. Gries e D. Zingaro, "Comparing outcomes in inverted and traditional CS1," em ITiCSE '14: Proceedings of the 2014 conference on Innovation & technology in computer science education, 2014.
- [30] J. Pattanaphanchai, "An Investigation of Students' Learning Achievement of Students' Learning Achievement and Perception using Flipped Classroom," Journal of University Teaching and Learning Practice, vol. 16, n° 5, 2019.
- [31] U. Schäfer, "Teaching Modern C++ with Flipped Classroom and Enjoyable IoT Hardware," em 2019 IEEE Global Engineering Education Conference (EDUCON), 2019.
- [32] L. Cao e M. Grabchak, "Interactive Preparatory Work in a Flipped Programming Course," em CompEd '19: Proceedings of the ACM Conference on Global Computing Education, 2019.
- [33] H. Hung, "Flipped Learning with Live-Coding Approach for Programming Concepts Learning," em 2018 1st International Cognitive Cities Conference (IC3), 2018.
- [34] S. Alhazbi, "Using Flipped Classroom Approach to Teach Computer Programming," em 2016 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE), 2016.
- [35] C. Latulipe, N. B. Long e C. E. Seminario, "Structuring Flipped Classes with Lightweight Teams and Gamification," em SIGCSE '15: Proceedings of the 46th ACM Technical Symposium on Computer Science Education, 2015.
- [36] S. Garcia, "Improving classroom preparedness using guided practice," em SIGCSE '18: Proceedings of the 49th ACM Technical Symposium on Computer Science Education, 2018.
- [37] A. Vihavainen, M. Luukkainen e J. Kurhila, "Multi-faceted support for MOOC in programming," em SIGITE '12: Proceedings of the 13th annual conference on Information technology education, 2012.
- [38] H. Özyurt e Ö. Özyurt, "Analyzing the effects of adapted flipped classroom approach on computer programming success, attitude toward programming, and programming self-efficacy," Comput. Appl. Eng. Education, vol. 26, p. 2036-2046, 2018.
- [39] B. Tyle e M. Abdrakhmanova, "Flipping the CS1 and CS2 classrooms in Central Asia," em 2016 IEEE Frontiers in Education Conference (FIE), 2016.
- [40] L. L. Lacher, A. Jiang, Y. Zhang e M. C. Lewis, "Including Coding Questions in Video Quizzes for a Flipped CS1," em SIGCSE '18: Proceedings of the 49th ACM Technical Symposium on Computer Science Education, 2018.
- [41] P. Seeling, "Switching to blend-Ed: Effects of replacing the textbook with the browser in an introductory computer programming course," em 2016 IEEE Frontiers in Education Conference (FIE), 2016.
- [42] J. Pattanaphanchai, "An investigation of students' learning achievement and perception using flipped classroom in an introductory programming course: A case study of Thailand higher education," Journal of University Teaching and Learning Practice, vol. 16, n° 4, 2019.
- [43] J. Gallaugh, S. Cho, J. FitzGibbon e J. A. Mahecha, "The Flipped Classroom and Mobile App Development: A New Model for Engaging Students in a First Programming Course," em AMCIS 2017 Proceedings, 2017.
- [44] L. L. Lacher e M. C. Lewis, "The effectiveness of video quizzes in a flipped class," em SIGCSE '15: Proceedings of the 46th ACM Technical Symposium on Computer Science Education, 2015.
- [45] Ö. Ö. Ö. Özyurt, "A qualitative study about enriching programming and algorithm teaching with flipped classroom approach," Pegem Journal of Education and Instruction, vol. 7, n° 2, pp. 189-210, 2017.
- [46] R. M. Clark, M. Besterfield-Sacre, D. Budny, K. M. Bursic, W. W. Clark, B. A. Norman, R. S. Parker, J. F. I. Patzer e W. S. Slaughter, "Flipping engineering courses: A school wide initiative," Advances in Engineering Education, vol. 5, n° 3, 2016.
- [47] D. Baldwin, "Can We 'Flip' Non-Major Programming Courses Yet?," em SIGCSE '15: Proceedings of the 46th ACM Technical Symposium on Computer Science Education, 2015.
- [48] H. Mok, "Teaching tip: The flipped classroom," Journal of Information Systems Education, vol. 25, n° 1, pp. 7-11, 2014.
- [49] C.-U. Lei, C.-W. Yau, K.-S. Lui, P. Yum, V. W. L. Tam, A. H. K. Yuen e E. Y. Lam, "Teaching Internet of Things: Enhancing learning efficiency via full-semester flipped classroom," em 2017 IEEE 6th International Conference on Teaching, Assessment, and Learning for Engineering (TALE), 2017.
- [50] Y. Hayashi, K. Fukamachi e H. Komatsugawa, "Collaborative learning in computer programming courses that adopted the flipped classroom," em 2015 International Conference on Learning and Teaching in Computing and Engineering, 2015.
- [51] A. Mohamed, "Designing a CS1 Programming Course for a Mixed-Ability Class," em WCCCE '19: Proceedings of the Western Canadian Conference on Computing Education, 2019.

- [52] J. R. d. Miras, J. R. Balsas-Almagro e Á. L. García-Fernández, “Using flipped classroom and peer instruction methodologies to improve introductory computer programming courses,” *Computer Applications in Engineering Education*, vol. 30, n° 2, pp. 133-145, 2022.
- [53] H. Loftsson e A. Matthíasdóttir, “Using Flipped Classroom and Team-Based Learning in a First-Semester Programming Course: An Experience Report,,” em 2019 IEEE International Conference on Engineering, Technology and Education (TALE), 2019.
- [54] A. G. d. O. Fassbinder, T. G. G. Botelho, R. J. Martins e E. F. Barbosa, “Applying flipped classroom and problem-based learning in a CS1 course,” em 2015 IEEE Frontiers in Education Conference (FIE), 2015.
- [55] S. R. Sobral e C. F. Oliveira, “Clustering Algorithm to Measure Student Assessment Accuracy,,” *Big Data Cogn. Computing*, vol. 51, n° 8, 2021.